
```
% Problem 5
% Constructing adjacency matrix A and graph G
n = 50;
n1 = 20;
n2 = 30;
A = zeros(n);
A(1:n1, 1:n1) = 1;
A(n1+1:n, n1+1:n) = 1;
A(n1+1:n1+3, 1:3) = eye(3);
A(1:3, n1+1:n1+3) = eye(3);
A = A - eye(n);
G = graph(A);

% Plot result
h = plot(G, 'NodeColor', 'b');
hold on;
title('Problem 5');

% Construct graph Laplacian
Dv = degree(G);
D = zeros(n);
for i = 1:n
    D(i, i) = Dv(i);
end
L = D - A;

% Find 2 potential solutions
a1 = (n1 - n2)/sqrt(n);
a2 = 2*sqrt(n1*n2/n);
[V,D] = eigs(L, 2, 'SA');
v2 = V(:, 2);
splus = a1 * ones(n, 1) + a2 * v2;
sminus = a1 * ones(n, 1) - a2 * v2;

sortedSPlus = sort(splus, 'descend');
sortedSMinus = sort(sminus, 'ascend');
maxValues = sortedSPlus(1:n1, 1);
minValues = sortedSMinus(1:n1, 1);
maxIds = zeros(0, 1);
minIds = zeros(0, 1);
for j = 1:n1
    if length(maxIds) < n1
        idMax = find(splus == maxValues(j, 1));
        if ~ismember(idMax, maxIds)
            maxIds = [maxIds; idMax];
        end
    else
        maxIds = maxIds(1:n1, 1);
    end
    if length(minIds) < n1
        idMin = find(sminus == minValues(j, 1));
        if ~ismember(idMin, minIds)
```

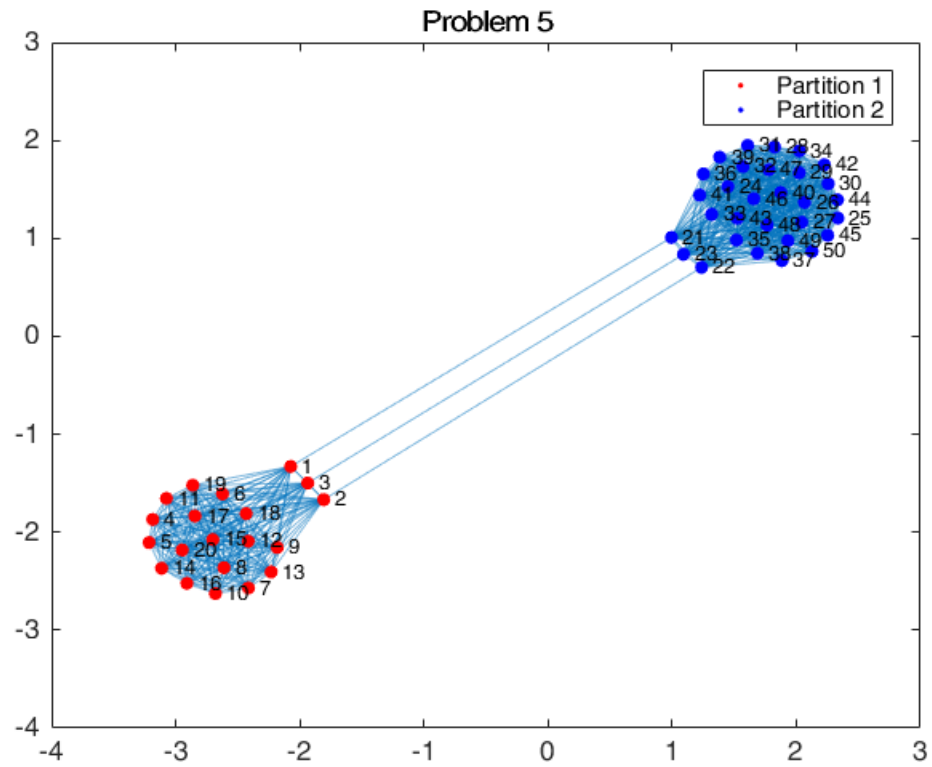
```

        minIds = [minIds; idMin];
    end
else
    minIds = minIds(1:n1, 1);
end
end
for i = 1:n
    if ismember(i, maxIds)
        splus(i) = 1;
    else
        splus(i) = -1;
    end
    if ismember(i, minIds)
        sminus(i) = 1;
    else
        sminus(i) = -1;
    end
end

% Calculate the cut size and pick solution with the smallest
splusCutSize = 1/4 * splus' * L * splus;
sminusCutSize = 1/4 * sminus' * L * sminus;
syms solution;
if splusCutSize < sminusCutSize
    solution = splus;
else
    solution = sminus;
end

% Highlight the relevant vertices on the graph
ids = find(solution == 1);
highlight(h, ids, 'NodeColor', 'r');

h = zeros(2, 1);
h(1) = plot(NaN,NaN,'.r');
h(2) = plot(NaN,NaN,'.b');
legend(h, 'Partition 1','Partition 2');
```



Published with MATLAB® R2016a